

A Basis for all Solutions of the Key Equation for Gabidulin Codes

Antonia Wachter, Vladimir Sidorenko and Martin Bossert

Institute of Telecommunications and Applied Information Theory

University of Ulm, Germany

{antonia.wachter | vladimir.sidorenko | martin.bossert}@uni-ulm.de

Abstract—We present and prove the correctness of an efficient algorithm that provides a basis for all solutions of a key equation in order to decode Gabidulin (\mathcal{G} -) codes up to a given radius τ . This algorithm is based on a symbolic equivalent of the *Euclidean Algorithm* (EA) and can be applied for decoding of \mathcal{G} -codes beyond half the minimum rank distance. If the key equation has a unique solution, our algorithm reduces to Gabidulin's decoding algorithm up to half the minimum distance. If the solution is not unique, we provide a basis for all solutions of the key equation. Our algorithm has time complexity $\mathcal{O}(\tau^2)$ and is a generalization of the modified EA by Bossert and Bezzateev for Reed-Solomon codes.

I. INTRODUCTION

A special class of rank-metric codes was introduced by Delsarte [1], Gabidulin [2] and Roth [3]. These codes are also called Gabidulin (\mathcal{G} -) codes. Kötter and Kschischang recently constructed network codes based on \mathcal{G} -codes [4].

In [2], Gabidulin presented an algorithm for decoding \mathcal{G} -codes up to half the minimum (rank) distance with a symbolic equivalent of the *Euclidean Algorithm* (EA). Paramonov and Tretjakov [5] and independently Richter and Plass [6], [7] gave a generalization of the *Berlekamp-Massey Algorithm* (BMA) for decoding of \mathcal{G} -codes up to half the minimum distance. This algorithm was proved and extended by Sidorenko *et al.* in [8]. This generalization of the BMA yields a basis for all solutions of the key equation for decoding of \mathcal{G} -codes up to a given radius, if there is no unique solution.

In this paper, we present an algorithm that provides a basis for all solutions of the key equation using the symbolic equivalent of the EA. Our algorithm is a generalization of the *Bossert-Bezzateev Algorithm* (BBA) from [9]. The BBA was applied for decoding of interleaved *Reed-Solomon* (RS) codes beyond half the minimum distance using the EA.

This paper is organized as follows: In Section II, we give the required definitions and state the problem. Section III provides the algorithm and in Section IV, we prove the correctness of the algorithm. The paper ends with a conclusion in Section V.

II. DEFINITIONS AND PROBLEM FORMULATION

A. Linearized Polynomials

\mathcal{G} -codes are defined by means of *linearized polynomials* (see e.g. [10]). Let q be a power of a prime and let us denote the

Frobenius q -power by:

$$x^{[i]} = x^{q^i}. \quad (1)$$

A linearized polynomial over \mathbb{F}_{q^m} is a polynomial of the form

$$L(x) = \sum_{i=0}^t l_i x^{[i]}, \quad (2)$$

with $l_i \in \mathbb{F}_{q^m}$. If the coefficient $l_t \neq 0$, we define the q -degree by $\deg_q L(x) = t$.

An important property of linearized polynomials for all $a, b \in \mathbb{F}_{q^m}$ and all $\beta_1, \beta_2 \in \mathbb{F}_q$ is:

$$L(\beta_1 a + \beta_2 b) = \beta_1 L(a) + \beta_2 L(b). \quad (3)$$

Consequently, any linear combination of roots of a linearized polynomial $L(x)$ is also a root of $L(x)$.

Let $F(x)$ and $G(x)$ be linearized polynomials over \mathbb{F}_{q^m} . The *symbolic product* of $F(x)$ and $G(x)$ is:

$$F(x) \otimes G(x) = F(G(x)). \quad (4)$$

If $\deg_q F(x) = t_F$ and $\deg_q G(x) = t_G$, then $\deg_q (F(x) \otimes G(x)) = t_F + t_G$. The symbolic product satisfies associativity and distributivity, but in general it is non-commutative, i.e.: $F(x) \otimes G(x) \neq G(x) \otimes F(x)$.

We call $G(x)$ a *right symbolic divisor* of $A(x)$, if $A(x) = F(x) \otimes G(x)$ for some linearized polynomial $F(x)$. These operations convert the set of all linearized polynomials into a non-commutative ring with the identity element $x^{[0]} = x$.

We define a *symbolic equivalent* of the *Extended Euclidean Algorithm* (SEEA). Let $R_{-1}(x) = B(x)$ and $R_0(x) = A(x)$ be two linearized polynomials with $\deg_q B(x) > \deg_q A(x)$. The *right symbolic greatest common divisor* (rsgcd) is calculated by the following recursion:

$$\begin{aligned} R_{-1}(x) &= Q_1(x) \otimes R_0(x) + R_1(x) \\ R_0(x) &= Q_2(x) \otimes R_1(x) + R_2(x) \\ &\vdots \\ R_{j-2}(x) &= Q_j(x) \otimes R_{j-1}(x) + R_j(x) \\ R_{j-1}(x) &= Q_{j+1}(x) \otimes R_j(x), \end{aligned} \quad (5)$$

where $\deg_q R_i(x) < \deg_q R_{i-1}(x)$. The last non-zero remainder $R_j(x)$ is the rsgcd($A(x), B(x)$).

Let $U_i(x)$ and $V_i(x)$ be polynomials, which can be calculated recursively:

$$\begin{aligned} U_i(x) &= -Q_i(x) \otimes U_{i-1}(x) + U_{i-2}(x) \\ V_i(x) &= -Q_i(x) \otimes V_{i-1}(x) + V_{i-2}(x), \end{aligned} \quad (6)$$

with $U_{-1}(x) = 0$, $U_0(x) = x^{[0]}$ and $V_{-1}(x) = x^{[0]}$, $V_0(x) = 0$. By means of these polynomials, we can write each remainder as a combination of the input polynomials $A(x)$ and $B(x)$:

$$R_i(x) = U_i(x) \otimes A(x) + V_i(x) \otimes B(x). \quad (7)$$

An important property of the polynomials from the SEEA is:

$$\deg_q U_i(x) + \deg_q R_{i-1}(x) = \deg_q B(x). \quad (8)$$

The proof of (8) is similar to the proof for the usual EA [11].

B. Gabidulin Codes and Their Key Equation

A \mathcal{G} -codeword is a vector $\mathbf{c} \in \mathbb{F}_{q^m}^n$, where n is the codeword length:

$$\mathbf{c} = (c_0, c_1, \dots, c_{n-1}). \quad (9)$$

This vector can be mapped on an $m \times n$ matrix \mathbf{C} with entries from \mathbb{F}_q . The *rank norm* $\text{rank}_q(\mathbf{c})$ is the rank of \mathbf{C} over \mathbb{F}_q .

For $n \leq m$, a linear (n, k) \mathcal{G} -code over \mathbb{F}_{q^m} is defined by its $(n - k) \times n$ parity check matrix \mathbf{H} (see [2]):

$$\mathbf{H} = \begin{pmatrix} h_1 & h_2 & \dots & h_n \\ h_1^{[1]} & h_2^{[1]} & \dots & h_n^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{[n-k-1]} & h_2^{[n-k-1]} & \dots & h_n^{[n-k-1]} \end{pmatrix}, \quad (10)$$

where $h_1, \dots, h_n \in \mathbb{F}_{q^m}$ are linearly independent over \mathbb{F}_q . The *minimum rank distance* d of the code \mathcal{G} is defined by:

$$d = \min\{\text{rank}_q(\mathbf{c}) \mid \mathbf{c} \in \mathcal{G}, \mathbf{c} \neq \mathbf{0}\}, \quad (11)$$

and is $d = n - k + 1$.

The transmitted codeword \mathbf{c} is corrupted by an additive error \mathbf{e} with $e_i \in \mathbb{F}_{q^m}$ of $\text{rank}_q(\mathbf{e}) = t$:

$$\mathbf{r} = \mathbf{c} + \mathbf{e}, \quad (12)$$

where \mathbf{r} is the received vector with $r_i \in \mathbb{F}_{q^m}$. We use a $t \times n$ matrix \mathbf{Y} of rank t with elements from \mathbb{F}_q to write:

$$\mathbf{e} = \mathbf{E} \cdot \mathbf{Y} = (E_1, E_2, \dots, E_t) \cdot \mathbf{Y}, \quad (13)$$

with E_1, E_2, \dots, E_t from \mathbb{F}_{q^m} are linearly independent over \mathbb{F}_q . The syndrome \mathbf{s} is calculated by:

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T = (S_0, S_1, \dots, S_{d-2}), \quad (14)$$

and can also be represented as a linearized polynomial:

$$S(x) = \sum_{i=0}^{d-2} S_i x^{[i]}. \quad (15)$$

The most important part of decoding \mathcal{G} -codes is solving the following key equation in order to find an *error span polynomial* $\Lambda(x)$ and an auxiliary polynomial $\Omega(x)$. $\Lambda(x)$ contains all linear combinations of E_1, E_2, \dots, E_t (13) as roots.

Theorem 1 [2] *Let the syndrome $S(x)$ and the minimum distance d be known, then the key equation for \mathcal{G} -codes is:*

$$\Omega(x) = \Lambda(x) \otimes S(x) \mod x^{[d-1]}, \quad (16)$$

where $\Omega(x)$ and $\Lambda(x)$ are linearized polynomials with the degree constraints:

$$\deg_q \Omega(x) < \deg_q \Lambda(x). \quad (17)$$

In this paper, we restrict ourselves to solving the key equation. The decoding procedure afterwards can be done as explained e.g. in [2]. In the following, let τ denote the decoding radius.

C. Solving the Key Equation up to Half the Minimum Distance

Gabidulin presented in [2] a method to solve the key equation using the SEEA if $\deg_q \Lambda(x) \leq \tau = \lfloor \frac{d-1}{2} \rfloor$. The Gabidulin algorithm is similar to the algorithm by Sugiyama *et al.* for RS codes [11] and is shown in Algorithm 1. In this procedure, the SEEA runs with the input polynomials $R_{-1}(x) = x^{[d-1]}$ and $R_0(x) = S(x)$ (Lines 1-5) until the degree constraints of Line 6 are fulfilled. Line 8 yields the solution to the key equation, where $a \in \mathbb{F}_{q^m}$ is an arbitrary constant factor. Often, a is chosen such that $\Lambda(x)$ is monic. Similar as for RS codes, it can be shown that the polynomials $\Lambda(x)$ and $\Omega(x)$ are unique except for the constant factor a .

Algorithm 1: Solving the Key Equation if $\tau = \lfloor \frac{d-1}{2} \rfloor$ [2]

Input: Syndrome $S(x)$, $x^{[d-1]}$
Initialize: $i \leftarrow 0$, $R_{-1}(x) \leftarrow x^{[d-1]}$, $R_0(x) \leftarrow S(x)$,
 $U_{-1}(x) \leftarrow 0$, $U_0(x) \leftarrow x^{[0]}$
1 while $R_i(x) \neq 0$ **do**
2 $i \leftarrow i + 1$
3 Calculate $Q_i(x)$ and $R_i(x)$ such that:
4 $R_{i-2}(x) = Q_i(x) \otimes R_{i-1}(x) + R_i(x)$
5 $U_i(x) \leftarrow -Q_i(x) \otimes U_{i-1}(x) + U_{i-2}(x)$
6 **if** $\deg_q R_{i-1}(x) \geq \lfloor \frac{d-1}{2} \rfloor$ **and** $\deg_q R_i(x) < \lfloor \frac{d-1}{2} \rfloor$ **then**
7 **break**
8 $\Lambda(x) \leftarrow a \cdot U_i(x)$ and $\Omega(x) \leftarrow a \cdot R_i(x)$
Output: $\Lambda(x)$, $\Omega(x)$

Note that a proper $\Lambda(x)$ has q -degree $t_\lambda = \deg_q \Lambda(x) \leq \tau$ and contains all linear combinations of t_λ linearly independent elements from \mathbb{F}_{q^m} as roots. In practice, if Algorithm 1 fails, decoding can be done with our algorithm that we provide in Section III.

D. Problem Formulation

The problem of finding all solutions of the key equation if $\tau > \lfloor \frac{d-1}{2} \rfloor$ can be formulated as follows.

Problem 1 *Let an integer τ , with $\lfloor \frac{d-1}{2} \rfloor < \tau < d - 1$, the syndrome $S(x)$ and $x^{[d-1]}$ be known. Assume, $S(x)$ results*

from an error vector \mathbf{e} with $\text{rank}_q(\mathbf{e}) \leq \tau$ (14). Find all pairs of polynomials $\{\Lambda(x), \Omega(x)\}$ with

$$\deg_q \Omega(x) < \deg_q \Lambda(x) = \tau, \quad (18)$$

such that the key equation (16) is fulfilled.

We want to solve this problem using the SEEA in an efficient way. The restriction $\text{rank}_q(\mathbf{e}) \leq \tau$ means that we limit the decoding radius to τ . We introduce τ_0 :

$$\tau = \left\lfloor \frac{d-1}{2} \right\rfloor + \tau_0. \quad (19)$$

Of course, the key equation (16) can be solved with standard methods (e.g. Gaussian elimination) with complexity $\mathcal{O}(\tau^4)$ operations in \mathbb{F}_{q^m} . However, we want an *efficient* solution.

In the following, we give an efficient algorithm to solve Problem 1 that has complexity $\mathcal{O}(\tau^2)$. The main results of our paper are Algorithm 2 and Theorem 2.

III. THE ALGORITHM FOR SOLVING THE KEY EQUATION

In this section, we give an efficient algorithm (Algorithm 2) based on the SEEA which solves Problem 1 with complexity $\mathcal{O}(\tau^2)$. We explain the different steps of Algorithm 2 in this section and give the proofs in Section IV.

Algorithm 2: Basis for all Solutions if $\tau > \left\lfloor \frac{d-1}{2} \right\rfloor$

Input: Syndrome $S(x)$, d , τ

Initialize: $i \leftarrow 0$, $j \leftarrow 0$, $R_{-1}(x) \leftarrow x^{[d-1]}$, $R_0(x) \leftarrow S(x)$,
 $U_{-1}(x) \leftarrow 0$, $U_0(x) \leftarrow x^{[0]}$,
 $\Delta_0(x) \leftarrow x^{[0]}$, $P_0(x) \leftarrow S(x)$

```

1 while  $R_i(x) \neq 0$  do
2    $i \leftarrow i + 1$ ,  $j \leftarrow j + 1$ 
3   Calculate  $Q_i(x)$  and  $R_i(x)$  such that:
4      $R_{i-2}(x) = Q_i(x) \otimes R_{i-1}(x) + R_i(x)$ 
5    $U_i(x) \leftarrow -Q_i(x) \otimes U_{i-1}(x) + U_{i-2}(x)$ 
6   while  $\deg_q U_i(x) - \deg_q \Delta_{j-1}(x) > 1$  do
7      $\Delta_j(x) \leftarrow x^{[1]} \otimes \Delta_{j-1}(x)$ 
8      $P_j(x) \leftarrow x^{[1]} \otimes P_{j-1}(x)$ 
9      $j \leftarrow j + 1$ 
10   $\Delta_j(x) \leftarrow U_i(x)$  and  $P_j(x) \leftarrow R_i(x)$ 
11 Calculate  $\Delta^{\mathcal{I}}, P^{\mathcal{I}}$  using (24), (25)
```

Output: $\Delta^{\mathcal{I}}, P^{\mathcal{I}}$

Algorithm 2 executes the SEEA (5) with the input polynomials $R_{-1}(x) = x^{[d-1]}$ and $R_0(x) = S(x)$. In each step i , the SEEA returns the remainder $R_i(x)$ and the polynomial $U_i(x)$ (6). This corresponds to Lines 1 until 5 of Algorithm 2.

The q -degree of the remainder $R_i(x)$ is decreasing in every step, but not necessarily by one. Equivalently, the q -degree of $U_i(x)$ is increasing in every step of the SEEA, but also not necessarily by one. Hence,

$$\begin{aligned} \deg_q R_{i+1}(x) &\leq \deg_q R_i(x) - 1, \\ \deg_q U_{i+1}(x) &\geq \deg_q U_i(x) + 1. \end{aligned} \quad (20)$$

The missing degrees are then filled up as given in Lines 6 until 9 of Algorithm 2. This definition assures that there exist polynomials $\Delta_i(x)$, $P_i(x)$ of each q -degree from 0 to $d-2$. This is shown in Lemma 1.

Lemma 1 *There exist polynomials of each degree from 0 to $d-2$ in each set $\{\Delta_i(x)\}$ and $\{P_i(x)\}$.*

Proof: At first assume that $\deg_q U_{i+1}(x) = \deg_q U_i(x) + 1$ and $\deg_q R_{i+1}(x) = \deg_q R_i(x) - 1$ for all i . Hence, all $U_i(x) = \Delta_i(x)$ and all $R_i(x) = P_i(x)$ have different degrees.

Now, consider the case when some degrees have to be filled up in Lines 6 until 9. Assume, that the q -degree of the $(i-1)$ th remainder decreases by more than one, i.e.:

$$\deg_q R_{i-1}(x) = \deg_q R_{i-2}(x) - \delta \quad \text{with } \delta > 1. \quad (21)$$

Since $\deg_q R_i(x) < \deg_q R_{i-1}(x)$, we know using Line 4 from Algorithm 2:

$$\deg_q Q_i(x) = \deg_q R_{i-2}(x) - \deg_q R_{i-1}(x) = \delta. \quad (22)$$

With the calculation of $U_i(x)$ (Line 5), we obtain:

$$\deg_q U_i(x) = \deg_q U_{i-1}(x) + \deg_q Q_i(x) = \deg_q U_{i-1}(x) + \delta. \quad (23)$$

Hence, if the degree of the remainders decreases by δ in step i , then the degree of $U_{i+1}(x)$ increases by δ in step $i+1$. If we multiply $R_i(x)$ symbolically from the left with $x^{[1]}$ and do this $(\delta-1)$ times, we fill up the $\delta-1$ missing degrees of the remainders. The same holds for the $U_i(x)$.

Due to (8), the q -degree of the last calculated $U_i(x)$ is $d-2$ and the q -degree of the last remainder is 0. Thus, all degrees from 0 to $d-2$ exist in the sets $\{\Delta_i(x)\}$ and $\{P_i(x)\}$. ■

If polynomials have different q -degrees, they are linearly independent. This becomes clear, if we write the coefficients of the polynomials as vectors. Consequently, with Lemma 1, the polynomials $\{\Delta_i(x)\}$ and $\{P_i(x)\}$ are linearly independent.

Figure 1 shows an example how the gaps are filled up. The upper half of this figure shows the q -degrees of $R_i(x)$, $U_i(x)$ and the lower half the q -degrees of $P_i(x)$, $\Delta_i(x)$. For example, there is a q -degree difference of 2 between $R_1(x)$ and $R_2(x)$. Consequently, in the third step, $\deg_q U_3(x) = \deg_q U_2(x) + 2$. The lower half of the picture shows that after filling up these q -degrees, all q -degrees from 0 to $d-2$ exist.

Subsequently, we define a subset of these polynomials (Line 11 of Algorithm 2):

$$\mathcal{I} = \{i \mid \deg_q \Delta_i(x) \leq \tau \wedge \deg_q P_i(x) < \tau\}. \quad (24)$$

In Lemma 2, we show that the set \mathcal{I} determines $2\tau_0 + 1$ pairs of polynomials $\{\Delta_i(x), P_i(x)\}$, which we denote by:

$$\Delta^{\mathcal{I}} = \{\Delta_i(x) \mid i \in \mathcal{I}\} \quad \text{and} \quad P^{\mathcal{I}} = \{P_i(x) \mid i \in \mathcal{I}\}. \quad (25)$$

The following linear combinations of polynomials provide all solutions of Problem 1 (see Theorem 2):

$$\Lambda(x) = \sum_i \beta_i \Delta_i^{\mathcal{I}}(x) \quad \text{and} \quad \Omega(x) = \sum_i \beta_i P_i^{\mathcal{I}}(x), \quad (26)$$

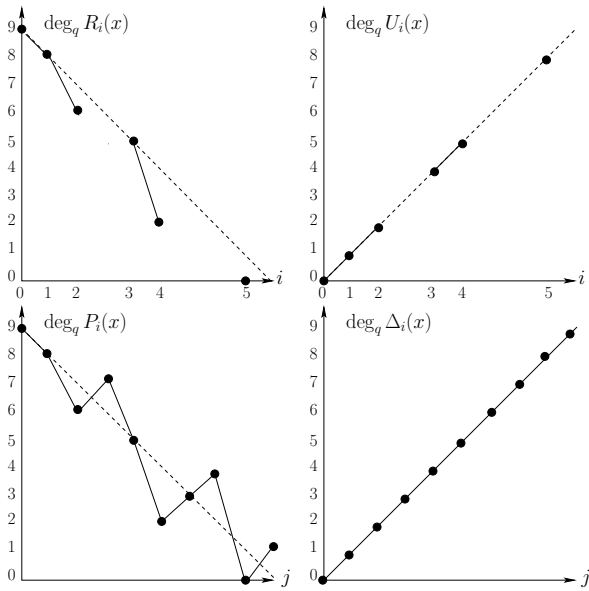


Fig. 1. Filling up the gaps in the degrees ($d = 11$)

with $\beta_i \in \mathbb{F}_{q^m}$. The definition of \mathcal{I} assures that $\deg_q \Lambda(x) = \tau$ and $\deg_q \Omega(x) < \tau$. Lemma 3 gives the proof that these linear combinations satisfy the key equation (16). In Theorem 2 we prove that there are only $(q^m)^{2\tau_0+1}$ solutions of Problem 1 and that these linear combinations provide all solutions.

Thus, Algorithm 2 provides a basis for all solutions of Problem 1. The algorithm only requires a complexity of $\mathcal{O}(\tau^2)$ operations in \mathbb{F}_{q^m} and is a generalization of the BBA [9]. The two algorithms are equivalent if $m = 1$, i.e. if $\mathbb{F}_{q^m} = \mathbb{F}_q$, since then $\alpha^{[i]} = \alpha$ for all elements from \mathbb{F}_q .

IV. PROOF OF THE CORRECTNESS OF THE ALGORITHM

In order to prove that Algorithm 2 solves Problem 1, we show with Lemmas 1 and 2 that each of the sets $\Delta^\mathcal{I}$ and $P^\mathcal{I}$ consists of $2\tau_0 + 1$ linearly independent polynomials. Afterwards, we prove that the linear combinations from (26) fulfill the key equation (Lemma 3) and that they give *all* solutions of Problem 1 (Theorem 2). Lemma 4 gives some properties that we need for the proof of Theorem 2.

Lemma 2 *The set $\mathcal{I} = \{i \mid \deg_q \Delta_i(x) \leq \tau \wedge \deg_q P_i(x) < \tau\}$ (24) has cardinality $2\tau_0 + 1$.*

Proof: At first assume, that $\deg_q U_{i+1}(x) = \deg_q U_i(x) + 1$ and $\deg_q R_{i+1}(x) = \deg_q R_i(x) - 1$ for all i . With (8):

$$\deg_q U_i(x) + \deg_q R_i(x) = \deg_q B(x) - 1 = d - 2. \quad (27)$$

The remainder $R_m(x)$ with $\deg_q R_m(x) = \tau - 1$ determines the smallest i in the set \mathcal{I} as $\deg_q R_i(x)$ is decreasing with increasing i . The polynomial $U_n(x)$ with $\deg_q U_n(x) = \tau$ determines the largest i as $\deg_q U_i(x)$ is increasing with i . If $\deg_q U_n(x) = \tau$, we know with (27) that $\deg_q R_n(x) = d - 2 - \tau$. Hence, with (19):

$$|\mathcal{I}| = \deg_q R_m(x) - \deg_q R_n(x) + 1 = 2\tau_0 + 1. \quad (28)$$

If some degrees have to be filled up, we know from Lemma 1 that the sets contain all degrees. Therefore, the number of polynomials in \mathcal{I} does not change, only the elements change and the cardinality is also $2\tau_0 + 1$. ■

Thus, with Lemma 1 and 2, each set $\Delta^\mathcal{I}$ and $P^\mathcal{I}$ consists of $2\tau_0 + 1$ linearly independent polynomials.

Lemma 3 *Let $\Delta^\mathcal{I}$ and $P^\mathcal{I}$ be the sets of polynomials calculated by Algorithm 2. Any pair of polynomials $\{\Lambda(x), \Omega(x)\}$ calculated by the linear combinations from (26) satisfies the key equation (16) with the degree constraints from (18).*

Proof: For each $R_i(x)$ and $U_i(x)$, the following holds:

$$R_i(x) = U_i(x) \otimes S(x) \mod x^{[d-1]}, \quad (29)$$

hence every polynomial that is a direct output of the SEEA satisfies the key equation. For the polynomials that are filled up as in Algorithm 2, we obtain:

$$x^{[k]} \otimes R_i(x) = x^{[k]} \otimes U_i(x) \otimes S(x) \mod x^{[d-1]}, \quad (30)$$

where k is the number of missing degrees between i and the current q -degree $j = i + k$. This is equivalent to:

$$\Delta_j(x) = P_j(x) \otimes S(x) \mod x^{[d-1]}. \quad (31)$$

Calculating the linear combinations from (26), we obtain due to the distributivity of the symbolic product:

$$\sum_i \beta_i \Delta_i^\mathcal{I}(x) = \left[\sum_i \beta_i P_i^\mathcal{I}(x) \right] \otimes S(x) \mod x^{[d-1]}, \quad (32)$$

which satisfies the key equation (16). Due to the definition of \mathcal{I} from (24), also the degree constraints from (18) are fulfilled. ■

Now, we rewrite the key equation (16) in order to prove that the linear combinations (26), provide *all* solutions of the key equation. $\Lambda(x)$ is a linearized polynomial with q -degree τ and hence has $\tau + 1$ unknown coefficients. Therefore,

$$\begin{aligned} \Omega(x) &= \Lambda(S(x)) \mod x^{[d-1]} \\ &= x^{[0]}(\Lambda_0 S_0) \\ &\quad + x^{[1]}(\Lambda_0 S_1 + \Lambda_1 S_0^{[1]}) \\ &\quad + \dots \\ &\quad + x^{[\tau]}(\Lambda_0 S_\tau + \Lambda_1 S_{\tau-1}^{[1]} + \dots + \Lambda_\tau S_0^{[\tau]}) \\ &\quad + x^{[\tau+1]}(\Lambda_0 S_{\tau+1} + \Lambda_1 S_\tau^{[1]} + \dots + \Lambda_\tau S_1^{[\tau]}) \\ &\quad + \dots \\ &\quad + x^{[d-2]}(\Lambda_0 S_{d-2} + \Lambda_1 S_{d-3}^{[1]} + \dots + \Lambda_\tau S_{d-\tau-2}^{[\tau]}). \end{aligned}$$

We claim in (18) that $\deg_q \Omega(x) < \tau$ and hence the coefficients of $x^{[\tau]}, x^{[\tau+1]}, \dots, x^{[d-2]}$ must be zero. Consequently, we have $d - 2 - \tau + 1$ equations which fix some Λ_i .

We have to check if these equations are linearly independent in order to know how many coefficients are fixed. If we rewrite this linear system of equations in matrix form, we obtain:

$$\mathbf{S} \cdot (\Lambda_\tau, \Lambda_{\tau-1}, \dots, \Lambda_0)^T = \mathbf{0}, \quad (33)$$

where \mathbf{S} is a $(d - \tau - 1) \times (\tau + 1)$ matrix:

$$\mathbf{S} = \begin{pmatrix} S_0^{[\tau]} & S_1^{[\tau-1]} & S_2^{[\tau-2]} & \dots & S_\tau \\ S_1^{[\tau]} & S_2^{[\tau-1]} & S_3^{[\tau-2]} & \dots & S_{\tau+1} \\ S_2^{[\tau]} & S_3^{[\tau-1]} & S_4^{[\tau-2]} & \dots & S_{\tau+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{d-\tau-2}^{[\tau]} & S_{d-\tau-1}^{[\tau-1]} & S_{d-\tau}^{[\tau-2]} & \dots & S_{d-2} \end{pmatrix}. \quad (34)$$

For this matrix, the following lemma holds.

Lemma 4 *Let \mathbf{S} (34) be the matrix of syndrome elements satisfying the requirements of Problem 1, then $\text{rank}(\mathbf{S}) = \min\{d - 1 - \tau, t = \text{rank}_q(\mathbf{e})\}$.*

Proof: We assume in Problem 1 that $t = \text{rank}_q(\mathbf{e}) \leq \tau$. With (13) and (14), we can rewrite the syndrome by (see [2]):

$$\mathbf{s} = \mathbf{E} \cdot \mathbf{Y} \cdot \mathbf{H}^T = \mathbf{E} \cdot \mathbf{X}, \quad (35)$$

where the elements of the $t \times (d - 1)$ -matrix \mathbf{X} in row i and column j are: $\mathbf{X}_{i,j} = x_i^{[j-1]}$ for $i = 1, \dots, t$ and $j = 1, \dots, d - 1$. The x_i are linearly independent over \mathbb{F}_q as shown in [2]. (35) yields:

$$S_l^{[k]} = \sum_{j=1}^t E_j^{[k]} x_j^{[k+l]}, \quad (36)$$

where k is an arbitrary integer and $l = 0, \dots, d - 2$. We can then decompose \mathbf{S} as follows:

$$\mathbf{S} = \hat{\mathbf{X}} \cdot \hat{\mathbf{E}} = \begin{pmatrix} x_1^{[\tau]} & x_2^{[\tau]} & \dots & x_t^{[\tau]} \\ x_1^{[\tau+1]} & x_2^{[\tau+1]} & \dots & x_t^{[\tau+1]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[d-2]} & x_2^{[d-2]} & \dots & x_t^{[d-2]} \end{pmatrix} \cdot \begin{pmatrix} E_1^{[\tau]} & E_1^{[\tau-1]} & E_1^{[\tau-2]} & \dots & E_1^{[0]} \\ E_2^{[\tau]} & E_2^{[\tau-1]} & E_2^{[\tau-2]} & \dots & E_2^{[0]} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E_t^{[\tau]} & E_t^{[\tau-1]} & E_t^{[\tau-2]} & \dots & E_t^{[0]} \end{pmatrix}, \quad (37)$$

where x_1, \dots, x_t and E_1, \dots, E_t are linearly independent over \mathbb{F}_q . Both matrices, $\hat{\mathbf{X}}$ and $\hat{\mathbf{E}}$, are linearized Vandermonde-like matrices. Such a matrix always has full rank [10]. Therefore, $\text{rank}(\hat{\mathbf{X}}) = \min\{d - \tau - 1, t\}$ and $\text{rank}(\hat{\mathbf{E}}) = \min\{t, \tau + 1\} = t$. Since the rows of $\hat{\mathbf{E}}$ are linearly independent,

$$\text{rank}(\mathbf{S}) = \min\{d - \tau - 1, t = \text{rank}_q(\mathbf{e})\}. \quad (38)$$

Theorem 2 *Let $\Delta^\mathcal{I}$ and $P^\mathcal{I}$ be the sets of $2\tau_0 + 1$ linearly independent polynomials calculated with Algorithm 2. Let $\lfloor \frac{d-1}{2} \rfloor < \tau < d - 1$ and $\text{rank}_q(\mathbf{e}) \leq \tau$ hold.*

Any pair of polynomials $\{\Lambda(x), \Omega(x)\}$ satisfying the key equation (16) with the degree constraints (18), can be calculated by the linear combinations of the polynomials from $\Delta^\mathcal{I}, P^\mathcal{I}$ given in (26). That means, the sets $\Delta^\mathcal{I}, P^\mathcal{I}$ are a basis for all solutions of the key equation (16).

Proof: In the following, we show that only $2\tau_0 + 1$ coefficients of $\Lambda(x)$ can be chosen arbitrarily, if (16) has to be fulfilled. Hence, the set of $(q^m)^{2\tau_0+1}$ different linear combinations from (26) constitutes the set of all possible $\Lambda(x)$.

With (33), some coefficients of $\Lambda(x)$ are fixed. Lemma 4 shows that $\text{rank}(\mathbf{S}) = \min\{d - 1 - \tau, t = \text{rank}_q(\mathbf{e})\}$.

If $\text{rank}(\mathbf{S}) < d - \tau - 1$, then $\text{rank}(\mathbf{S}) = \text{rank}_q(\mathbf{e}) < d - \tau - 1$. We assume in the theorem $\tau > \lfloor \frac{d-1}{2} \rfloor$, hence this is equivalent to $\text{rank}_q(\mathbf{e}) < \lfloor \frac{d-1}{2} \rfloor$ and this error can always be corrected by Algorithm 1.

Otherwise, $\text{rank}(\mathbf{S}) = d - 1 - \tau$ and that means that $d - 1 - \tau$ coefficients of $\Lambda(x)$ are fixed. Since the number of coefficients of $\Lambda(x)$ is $\tau + 1$, the number of free coefficients is:

$$\tau + 1 - (d - 1 - \tau) = (2\tau - (d - 1)) + 1 = 2\tau_0 + 1. \quad (39)$$

Hence, there are only $(q^m)^{2\tau_0+1}$ possible $\Lambda(x)$. They can be calculated by the $(q^m)^{2\tau_0+1}$ linear combinations from (26) as the linear combinations satisfy the key equation and fulfill the degree constraints. Equivalently, all $\Omega(x)$ can be calculated by the linear combinations from (26). ■

The proof of Theorem 2 is one of the main results of this paper as it can be done in a similar way for RS codes. This proof is more descriptively than the proof given in [9] for RS codes.

V. CONCLUSION

In this paper, we have presented an efficient algorithm that provides a basis for all solutions of the key equation for decoding of \mathcal{G} -codes up to a certain radius τ . This algorithm requires $\mathcal{O}(\tau^2)$ operations in \mathbb{F}_{q^m} and can be applied for decoding of \mathcal{G} -codes beyond half the minimum rank distance. Our algorithm is based on a symbolic equivalent of the EA and is a generalization of the BBA.

REFERENCES

- [1] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Comb. Theory, Ser. A*, vol. 25, no. 3, pp. 226–241, 1978.
- [2] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Peredachi Inf.*, vol. 21, no. 1, pp. 3–16, 1985.
- [3] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 328–336, 1991.
- [4] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *Information Theory, IEEE Transactions on*, vol. 54, no. 8, pp. 3579–3591, 2008.
- [5] A. V. Paramonov and O. V. Tretjakov, "An analogue of Berlekamp-Massey algorithm for decoding codes in rank metric," in *MIPT*, 1991.
- [6] G. Richter and S. Plass, "Fast decoding of rank-codes with rank errors and column erasures," in *Proceedings. International Symposium on Information Theory, 2004. ISIT 2004*, 2004, p. 398.
- [7] —, "Error and Erasure Decoding of Rank-Codes with a Modified Berlekamp-Massey Algorithm," in *ITG Conference on Source and Channel Coding (SCC)*, 2004, 2004.
- [8] V. R. Sidorenko and M. Bossert, "Synthesizing all linearized shift-registers of the minimal or required length," in *ITG Conf. on Source and Channel Coding*, January 2010.
- [9] M. Bossert and S. Bezzateev, "Decoding of interleaved RS codes with the Euclidean algorithm," in *IEEE International Symposium on Information Theory, 2008. ISIT 2008*, 2008, pp. 1803–1807.
- [10] R. Lidl and H. Niederreiter, *Finite Fields*, ser. Enc. of Mathematics and its Applications. Cambridge University Press, October 1996.
- [11] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," *Information and Control*, vol. 27, no. 1, pp. 87–99, 1975.